

The True Destination of EGO is Multi-local Optimization

Simon Wessing¹, Mike Preuss²

¹Computer Science Department

Technische Universität Dortmund, Germany

²European Research Center for Information Systems

WWU Münster, Germany

`simon.wessing@tu-dortmund.de`, `mike.preuss@uni-muenster.de`

Efficient global optimization is a popular algorithm for the optimization of expensive multimodal black-box functions. One important reason for its popularity is its theoretical foundation of global convergence. However, as the budgets in expensive optimization are very small, the asymptotic properties only play a minor role and the algorithm sometimes comes off badly in experimental comparisons. Many alternative variants have therefore been proposed over the years. In this work, we show experimentally that the algorithm instead has its strength in a setting where multiple optima are to be identified.

1 Introduction

Efficient global optimization (EGO) is a popular algorithm for the optimization of expensive multimodal black-box functions. At its core is a Kriging metamodel, whose predictions are used to formulate a so-called infill criterion. This criterion usually is a compromise between two goals: a) to detect especially good solutions, and b) to improve the model itself in order to enable better predictions. The established model can be employed to cheaply search for potential new points because it is much faster than the original function, often by a factor of 1000 or more. By optimizing with regard to the infill criterion, a new point can be determined for sampling the expensive function. It is clear that this works reasonably well only for functions that can be predicted from a sparse sample, namely relatively low dimensional and generally rather smooth ones. With more and more samples coming in, the model improves, so that one gets a better and better overall impression of the original function. However, the model fit also gets

more and more expensive, due to necessary matrix inversions. Thus, the sample size is limited, usually to around 1000 points.

The infill criterion of choice for EGO is the so-called expected improvement (EI), which incorporates both objectives mentioned above. While EGO is conceptually elegant and its convergence rate to the global optimum can be analyzed mathematically [3], there are multiple experimental results that indicate a preference for a more greedy infill criterion in expensive global optimization. For example, Sóbester et al. [21] propose a weighted expected improvement, which can be adjusted to search more locally or globally, depending on the weights. Some more evidence has surfaced in research on “multipoint” infill criteria for parallelizing function evaluations. Bischl et al. [2] discovered that just using the model prediction was the most successful infill criterion in a comparison with expected improvement and several multipoint infill criteria for a budget of $45n$ objective function evaluations, where n is the number of decision variables of the problem. Ginsbourger et al. [10] showed that an explorative variant of their *constant liar* criterion was less competitive than the more exploitative one, when filling in four to ten points on the Branin function. Also Ursem [27] developed an *investment portfolio improvement* function to propose three solutions per iteration, ranging from high exploitation to high exploration.

Our position is, that while the rather global search strategy of expected improvement may prevent a highly accurate approximation of the global optimum with small budgets, it represents a virtue for the task of finding multiple optima. This task is also known under the names of multi-global or multi-local optimization, depending on if only global or all optima are sought. EGO has to store the sampled points and function values anyway, to build the model, and all we have to do is to add a basin identification heuristic, to decide which points correspond to distinct attraction basins, and simply select the best one of each basin. As we want to avoid the responsibility of deciding if an optimum is global or local in our algorithm, we focus on multi-local optimization here.

Employing EGO as multi-local optimization algorithm may seem counterintuitive at first, but we claim that in an expensive optimization scenario, where we can afford only several hundreds of function evaluations, this makes a lot of sense. Many competing multi-local optimization algorithms rely on (multiple) local searches which turn out to be too expensive in this scenario. The focus of this paper is to experimentally analyse how well EGO is suited for expensive multi-local optimization. As expensive optimization setting, we assume budgets of 500 evaluations or less here. With respect to the limitation concerning the number of points that can be used as basis of a Kriging model, EGO matches the expensive optimization scenario very well. We therefore run EGO against one of the most effective and robust black-box optimization methods, the (Restart-) CMA-ES [1]. The CMA-ES is a good reference as it is also part of modern multimodal optimization algorithms, e. g., it performs the local optimization step in the NEA2 method [16]. Feliot et al. [8] also compared model-based approaches with local search algorithms in a constrained optimization setting and found that the competition is quite balanced. This already shows that the situation is not as clear-cut as it seems.

Our comparison is carried out on a mix of typical global optimization benchmark functions and niching competition benchmark functions, and we look at the results from

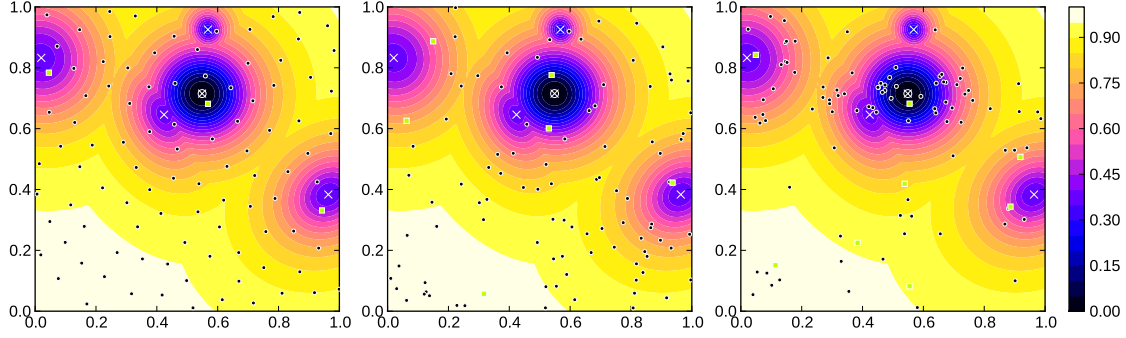


Figure 1: Examples of nearest-better clustering on a 2-D landscape, applied to $N = 100$ highly uniform, random uniform, and clustered points (from left to right). Selected points representing basin centers are marked with green squares.

two perspectives:

1. global optimization: we are only interested in locating the global optimum (one global optimizer in case it has several preimages),
2. multi-local optimization: we want to detect as many local/global optima as possible, ideally all of them.

If our reasoning from above is correct, EGO should perform better than CMA-ES and related methods under a multi-local optimization perspective, but worse if seen from a global optimization perspective. This would mean that in the expensive black-box setting, EGO is a very suitable multi-local optimization method. In order to compare it to other such methods, we need to add a basin identification method, because in the answer set of the algorithm, we only want points that are approximations of existing optima, not the full archive. According to [29], we have two methods at hand: topographical selection (TS) by [24] and nearest-better clustering (NBC) as described in [15].

Previous investigations showed that using NBC can be problematic because it produces too many clusters if the point set deviates from uniformity. An example for this effect can be found in Figure 1, where the uniformity decreases from left to right. If, e.g., the clustered points reside on a linear slope, the best point of a cluster is wrongly interpreted as representing an optimum where none exists (see rightmost subfigure, bottom left corner). The reason for this behavior is that outliers tend to be selected simply because of their large nearest-neighbor distances. We therefore rely on TS in this work.

To our knowledge, EGO and related surrogate-model-based methods have never been investigated in this way. Also the whole area of expensive multi-local/multimodal optimization seems to have been sparsely visited. One of the few works in this area is [14], however, they do not employ the term *multimodal optimization* as it has been established only years later.

The next section introduces necessary problem definitions, Sect. 3 explains the employed methods, namely EGO and topographical selection. The remainder of the paper

is mostly concerned with the comparison experiment (Sect. 4) and ends with the conclusions.

2 Problem definition

In the following, we will assume to have a deterministic objective function $f : \mathcal{X} \rightarrow \mathbb{R}$, where $\mathcal{X} = [\boldsymbol{\ell}, \boldsymbol{u}] \subset \mathbb{R}^n$ is the *search space* or *region of interest* (ROI) and $n \in \mathbb{N}$ is the fixed number of decision variables. The vectors $\boldsymbol{\ell} = (\ell_1, \dots, \ell_n)^\top$ and $\boldsymbol{u} = (u_1, \dots, u_n)^\top$ are called the lower and upper bounds of \mathcal{X} , respectively. Let $N(\boldsymbol{y}) = \{\boldsymbol{x} \in \mathcal{X} \mid d(\boldsymbol{x}, \boldsymbol{y}) \leq \epsilon\}$ be the neighborhood of a point $\boldsymbol{y} \in \mathcal{X}$. We say $f^* := f(\boldsymbol{x}^*)$ is a local minimum if $\exists \epsilon > 0 : \nexists \boldsymbol{x} \in N(\boldsymbol{x}^*) : f(\boldsymbol{x}) < f(\boldsymbol{x}^*)$. Technically, this implies that also plateaus are considered as local optima, although they are rather not intuitively, but at least this ensures that every position of a global optimum is also one of a local optimum. A multimodal optimization problem was implicitly already defined by Törn and Žilinskas [25, pp. 2–3]. In [28, p. 6], the definition was formulated explicitly as follows:

Definition 1 (Multimodal minimization problem). *Let there be ν local minima f_1^*, \dots, f_ν^* of f in \mathcal{X} . If the ordering of these optima is $f_{(1)}^* < \dots < f_{(l)}^* < h < \dots < f_{(\nu)}^*$, a multimodal minimization problem is given as the task to approximate the set $\bigcup_{i=1}^l X_{(i)}^*$, where $X_{(i)}^* = \{\boldsymbol{x} \in \mathcal{X} \mid f(\boldsymbol{x}) = f_{(i)}^*\}$.*

The variable h in this definition is simply a threshold to potentially exclude some of the worse optima. If $h = \infty$, we will be interested in all local optima of a problem. If $f_{(1)}^* < h < f_{(2)}^*$, we are only interested in approximating all the global optima. Let P be the obtained approximation set. Additional constraints may be applied to this set, to obtain more specific problem definitions. For example, the cardinality of P could be restricted by requiring $|P| \leq k$. If $k = 1$, we have the conventional global optimization problem, where typically only one solution is sought. Another issue are diversity requirements, which could be formulated by demanding $\forall \boldsymbol{x}, \boldsymbol{y} \in P, \boldsymbol{x} \neq \boldsymbol{y} : d(\boldsymbol{x}, \boldsymbol{y}) > \epsilon$, i. e., the distance between any two solutions may not be smaller than some threshold ϵ . Alternatively, also more sophisticated diversity measures on the set P may be calculated, maybe leading to a multiobjective formulation of the problem [18]. However, we will stick to the basic task of finding all optima ($h = \infty$) here.

3 Methods

Algorithm 1 illustrates the general sequential model-based optimization (MBO) framework, of which EGO is an instantiation. The main idea in model-based optimization is to approximate the expensive function $f(\boldsymbol{x})$ in every iteration by a regression model, which is much cheaper to evaluate. This is also called a meta-model or surrogate. We are using a Kriging model, which not only provides a direct estimation $\hat{f}(\boldsymbol{x})$ of the true function value $f(\boldsymbol{x})$ but also an estimation of the prediction standard error $\hat{s}(\boldsymbol{x})$, also called a local uncertainty measure. Our Kriging implementation follows the “empirical

Algorithm 1 Sequential model-based optimization

```
1: generate an initial design  $D \subset \mathcal{X}$ 
2:  $\mathbf{y} \leftarrow f(D)$ 
3: while total evaluation budget is not exceeded do
4:   fit surrogate on  $D$  and obtain  $\hat{f}, \hat{s}$ 
5:   get new design point  $\mathbf{x}'$  by optimizing the infill criterion based on  $\hat{f}, \hat{s}$ 
6:    $\mathbf{y}' \leftarrow f(\mathbf{x}')$  // evaluate new point
7:    $D \leftarrow (D, \mathbf{x}')$  // update design
8:    $\mathbf{y} \leftarrow (\mathbf{y}, \mathbf{y}')$  // update responses
9: end while
10: return  $\hat{\mathbf{y}}^* = \min(\mathbf{y})$  and the associated  $\hat{\mathbf{x}}^*$ 
```

Bayes” approach with a correlation kernel and a maximum likelihood estimation of its parameters [11].

The whole MBO concept has roots in response surface methodology, which was originally applied to physical experiments (with a human in the loop) [5, pp. 7–11]. It starts by exploring the parameter space with an initial design, often constructed in a space-filling fashion. The main sequential loop can be divided into two alternating stages:

1. Fit a response surface to training data (including estimation of the model’s parameters).
2. Use the surface to compute new search points under the assumption that the parameters are correct.

In [12], the now standard expected improvement criterion was proposed. It is defined as

$$\text{EI}(\mathbf{x}) = \mathbb{E}[\max\{0, \hat{\mathbf{y}}^* - \hat{f}(\mathbf{x})\}] = \left(\hat{\mathbf{y}}^* - \hat{f}(\mathbf{x})\right) \Phi\left(\frac{\hat{\mathbf{y}}^* - \hat{f}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) + \hat{s}(\mathbf{x}) \phi\left(\frac{\hat{\mathbf{y}}^* - \hat{f}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right),$$

where ϕ and Φ are the density and cumulative distribution function of the standard normal distribution, respectively. Hence, the sought point is $\mathbf{x}^* = \arg \max_{\mathbf{x} \in \mathcal{X}} \text{EI}(\mathbf{x})$.

Topographical selection is provided as pseudo-code in algorithm 2. It has similarities to nearest-better clustering because the basic idea is to compare points regarding their quality to their closest neighbors. While NBC argues with relative distances, TS relies on fixed-size k -neighborhoods, such that k is a parameter of the algorithm. We start with an empty graph and for every point, and detect the k nearest neighbors. For each neighbor, we add an edge that points from the worse to the better point. After finishing the loop, we return all points that have only incoming edges as basin representatives.

As performance measure for an approximation set P in the case of global optimization, we use the deviation from the global optimum $f_{\Delta} = \hat{\mathbf{y}}^* - f_1^*$. In the multi-local case the number of found optima $o = |\{\mathbf{x}^* \in X^* \mid d_{\text{nn}}(\mathbf{x}^*, P) \leq r\}|$ divided by the total number of optima $|X^*| = \nu$ as peak ratio $\text{PR}(P) = o/\nu$ is employed [26, 23]. Another measure

Algorithm 2 Topographical selection

Input: points $\mathcal{P} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, number k of nearest neighbors

Output: nodes of the topograph with no outgoing edges

```
1: create a directed graph  $G = (V, E)$  with  $V = \{v_1, \dots, v_N\}$  and  $E = \emptyset$ 
2: for all  $i \in \{1, \dots, N\}$  do
3:    $J \leftarrow$  indices of the  $k$  nearest neighbors of  $\mathbf{x}_i$ 
4:   for all  $j \in J$  do
5:     if  $f(\mathbf{x}_j) < f(\mathbf{x}_i)$  then
6:        $E \leftarrow E \cup \{(v_i, v_j)\}$  // add edge to graph
7:     else if  $f(\mathbf{x}_i) < f(\mathbf{x}_j)$  then
8:        $E \leftarrow E \cup \{(v_j, v_i)\}$  // add edge to graph
9:     end if
10:  end for
11: end for
12: return  $\{v \in V \mid \deg^+(v) = 0\}$  // select nodes with no outgoing edges
```

Table 1: Different magnitudes for the number of function evaluations N_f .

Magnitude	Application
$n \cdot 10^1$	Initial designs in model-based optimization [12]
$n \cdot 10^2$	Expensive optimization
$n \cdot 10^3$	
$n \cdot 10^4$	Budget of the CEC 2005 competition [22]
$n \cdot 10^5$	Budget of the CEC 2013 niching competition [13]
$n \cdot 10^6$	Budget of the black-box optimization benchmark (BBOB) [9]

is the averaged Hausdorff distance (AHD) [20]

$$\text{AHD}(P) = \max \left\{ \left(\frac{1}{\nu} \sum_{i=1}^{\nu} d_{\text{nn}}(\mathbf{x}_i^*, P)^p \right)^{1/p}, \left(\frac{1}{N} \sum_{i=1}^N d_{\text{nn}}(\mathbf{x}_i, X^*)^p \right)^{1/p} \right\},$$

by using X^* as a reference set. The function $d_{\text{nn}}(\mathbf{x}, X)$ denotes the Euclidean distance of a point \mathbf{x} to its nearest neighbor in a set of points X .

4 Experiment

Research question: How does the assessment of optimization algorithms depend on the performance measurement in expensive optimization, i. e., are the results in global optimization different from multi-local optimization?

Pre-experimental planning: Table 1 shows how some budgets are associated with research areas and benchmarks. Measuring consumed resources simply as the number of objective function evaluations is generally deemed admissible if this number is small, because then the assumption of expensive function evaluations in relation to the overhead of an

Table 2: Test problems used in the experiment.

Problem name	Dim. n	#Local optima	#Global optima	Ref.
Shekel5	4	5	1	[6]
Shekel7	4	7	1	[6]
Shekel10	4	10	1	[6]
Hartman3	3	3	1	[6]
Hartman6	6	2	1	[6]
Goldstein-Price	2	5	1	[6]
Branin	2	3	3	[6]
Vincent	2	36	36	[13]
Vincent	3	216	216	[13]
Modified Rastrigin	4	48	1	[4]
Modified Rastrigin	8	48	1	[4]
Six-hump camelback	2	6	2	[13]

optimization algorithm holds. For extremely large budgets, this is rather unlikely [7]. However, in expensive optimization, often very computationally demanding algorithms are used, so it is also an interesting question where the actual break-even point between two optimization algorithms is in terms of wall clock time.

Task: We assume an anytime scenario for assessment, that is, the algorithms could be stopped at any time. We record three different performance measures over the course of optimization, namely the deviation from the global optimum, the peak ratio (PR), and the averaged Hausdorff distance (AHD). For PR, the position of an optimum is considered as approximated if a point is within a Euclidean distance of 0.01 in the normalized search space (see setup below). AHD is used with an exponent of one. For PR and AHD, the solutions up to the measuring point are filtered by topographical selection (TS), to stay close to a real-world scenario. Topographical selection, originally proposed by [24], contains a parameter k , specifying a number of neighbors. To determine this parameter, we use the model

$$k(n, N) = 0.215n + 0.74N^{1/2},$$

depending on the dimension n and the number of points N . It was developed in [29] for random uniform samples. Although the solution sets produced by the optimization algorithms are not uniformly distributed (except for MmLHS, see below), we feel certain that this is not a severe problem, as TS proved quite robust to changes in the distribution in previous experiments [29].

Setup: Table 2 contains the test problems used in this experiment. They consist of the classic test set for global optimization by Dixon and Szegö [6], and some problems taken from the 2013 niching competition [13]. The former problems mostly contain fewer minima than the latter ones. However, the latter ones in part have other properties that make them easier, i. e., separability (modified Rastrigin) or no local optima (Vincent). All problems have in common the rather low dimension, bound constraints, and the fact

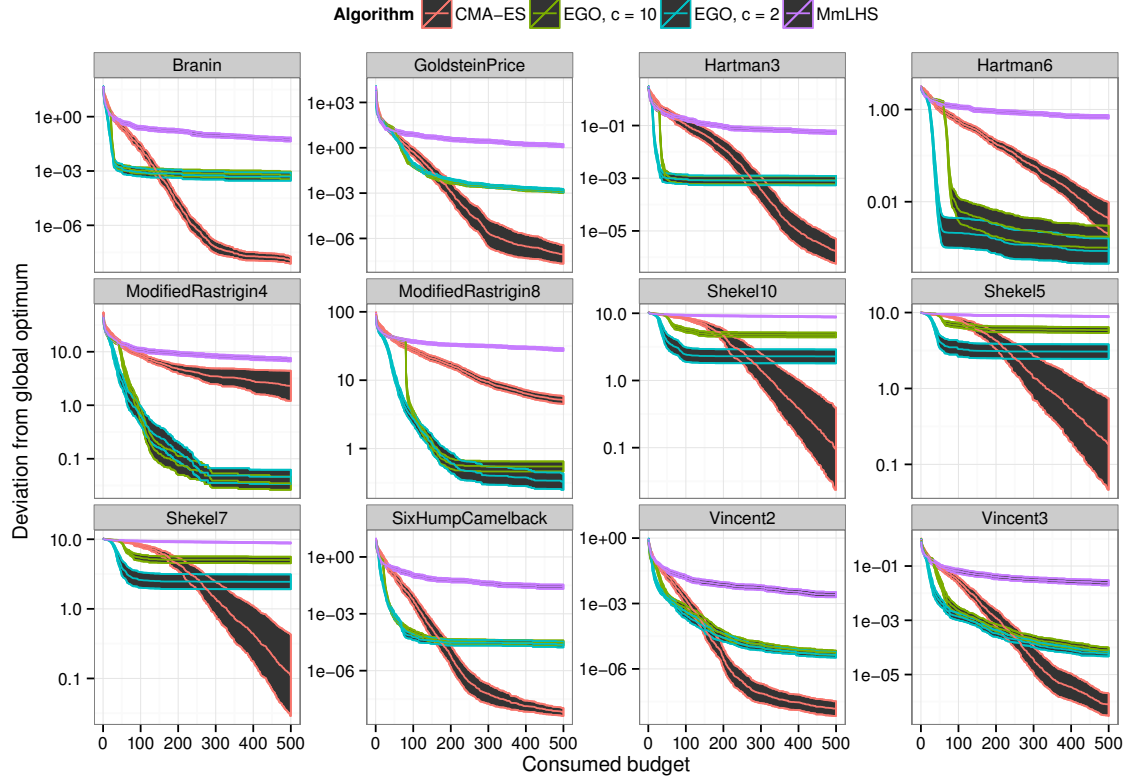


Figure 2: Convergence graphs for approximating the global optimum. Note the logarithmic scale.

that positions of *all* local and global minima are known. The last aspect is crucial for carrying out the assessment in the multi-local case with PR and AHD. The search spaces are always normalized to the unit hypercube.

We compare three different algorithms, namely CMA-ES, EGO, and a maximin Latin hypercube sampling (MmLHS). MmLHS acts as a representative of random search here. Our MmLHS designs are produced on-the-fly by a greedy construction heuristic. Thus, they are not exactly optimal according to the maximin-distance criterion, but possess a significantly higher uniformity than random uniform sampling (see [28, p. 58] for details). For EGO, we try two variants, which only differ in the amount of function evaluations invested into the initial sample. The sample size is determined as cn , with $c = 2$ or $c = 10$, in accordance with Tab. 1. The initial sample for EGO is also drawn by MmLHS, thus MmLHS alone can be seen as a limiting case for EGO where the whole budget is spent on the initial sample. EGO’s Kriging model uses the power-exponential kernel

$$\text{corr}(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(- \sum_{\ell=1}^n \theta_{\ell} |x_{i,\ell} - x_{j,\ell}|^p \right).$$

For the kernel parameters, we require $-2 \leq \log_{10}(\theta_{\ell}) \leq 2$ and $0.5 \leq p \leq 2$. As virtually all contemporary EGO implementations, our code differs from the algorithm in [12] in

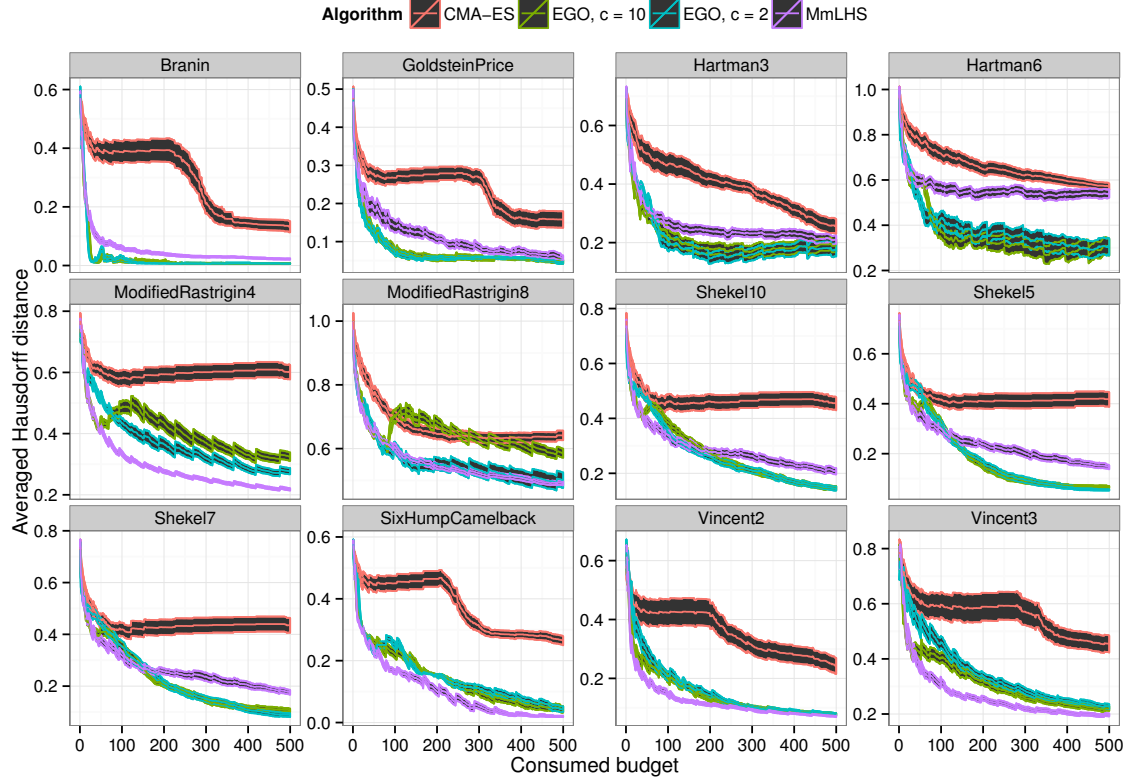


Figure 3: Averaged Hausdorff distances (AHD) over the course of optimization.

the way the infill criterion is optimized. Instead of a branch-and-bound approach, which consumes a lot of memory and restricts the kernel choice, we simply use CMA-ES, started once from the best of $100n$ uniformly distributed points. Also the likelihood function for fitting the model is optimized with CMA-ES, based on recommendations in [17].

CMA-ES is the candidate in this test set with a strong focus on local search. We use version 1.1.7 of the Python implementation¹. Its “tolfun” and “tolfunhist” stopping criteria are set to 10^{-3} and 10^{-5} , respectively, to stop really early and thus potentially have some budget left for starting another search. The starting points for CMA-ES are drawn by the maximin reconstruction algorithm, as recommended in [28]. The initial step size is set to 0.15.

In total, this experimental setup is chosen deliberately rather in favor of EGO than of CMA-ES, by including the test problems EGO was originally proposed for [12]. Additionally, CMA-ES is geared to being a very *robust* black-box optimizer, so it is not necessarily the most efficient one on these low-dimensional, continuously differentiable problems [19].

Results: Figures 2, 3, and 4 illustrate the development of the three indicators over the course of optimization. Additionally, we report the number of selected solutions in

¹<https://pypi.python.org/pypi/cma/>

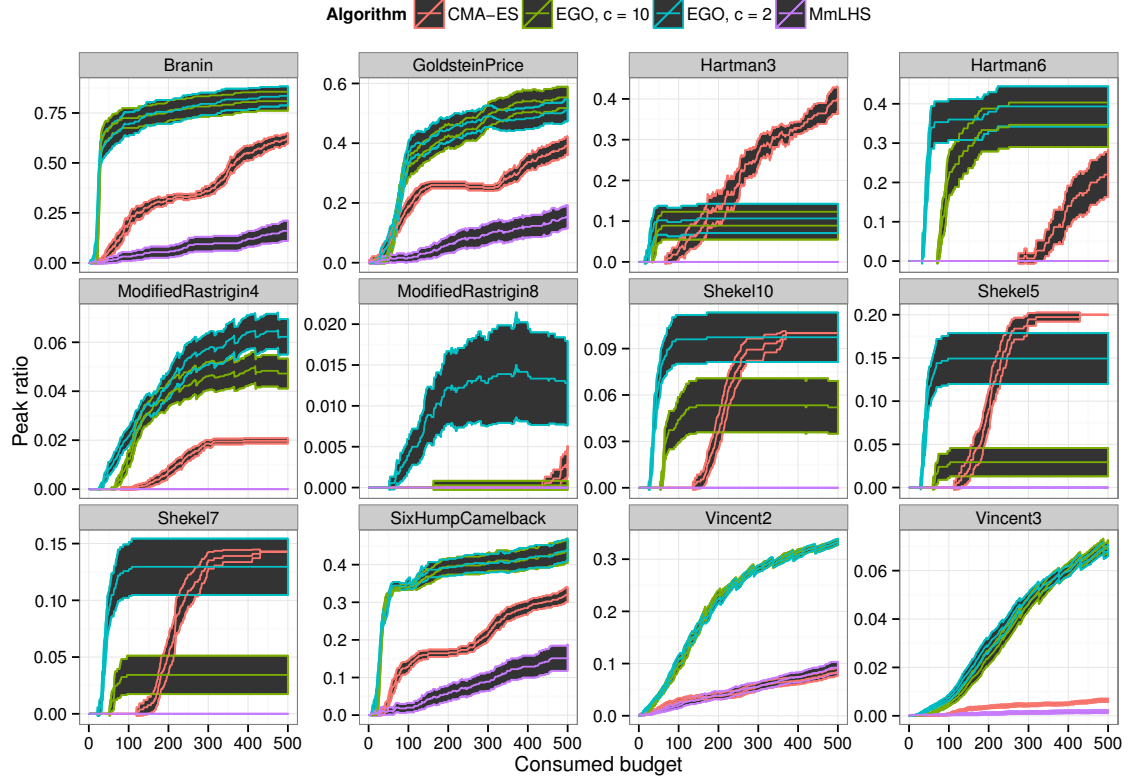


Figure 4: Peak ratio (PR, $r = 0.01$) over the course of optimization.

Fig. 5. Figure 6 shows the wall clock times of the algorithms. The curves contain the time for running the algorithm for the number of evaluations specified on the x-axis, plus the time for executing topographical selection once. In all figures, the curves represent mean values over 75 stochastic replications, with a 95% confidence interval for the mean under normality assumption.

Observations: Figure 2 shows that even under the quite restricted budget of 500 evaluations, CMA-ES significantly beats EGO on some problems EGO was developed for, if only the deviation from one global optimum counts. The variance is larger for CMA-ES, because some runs naturally only converge to local optima, but the average performance is clearly better. However, EGO is always better than CMA-ES in terms of AHD and PR. On problems with a large number of optima, MmLHS obtains a still better AHD than EGO, but the optima are not approximated very well. Thus, EGO always has the better peak ratio. With two exceptions, its peak ratio is also always better than that of CMA-ES.

Only few significant differences can be found between the choices $c = 2$ and $c = 10$ for EGO, but the results seem to be slightly in favor of $c = 2$.

Figure 5 shows that the number of selected solutions is mostly nicely correlated with the problems' actual number of optima. On Branin and Shekel5, even the correct number of optima is reliably identified towards the end of the optimization. However, the

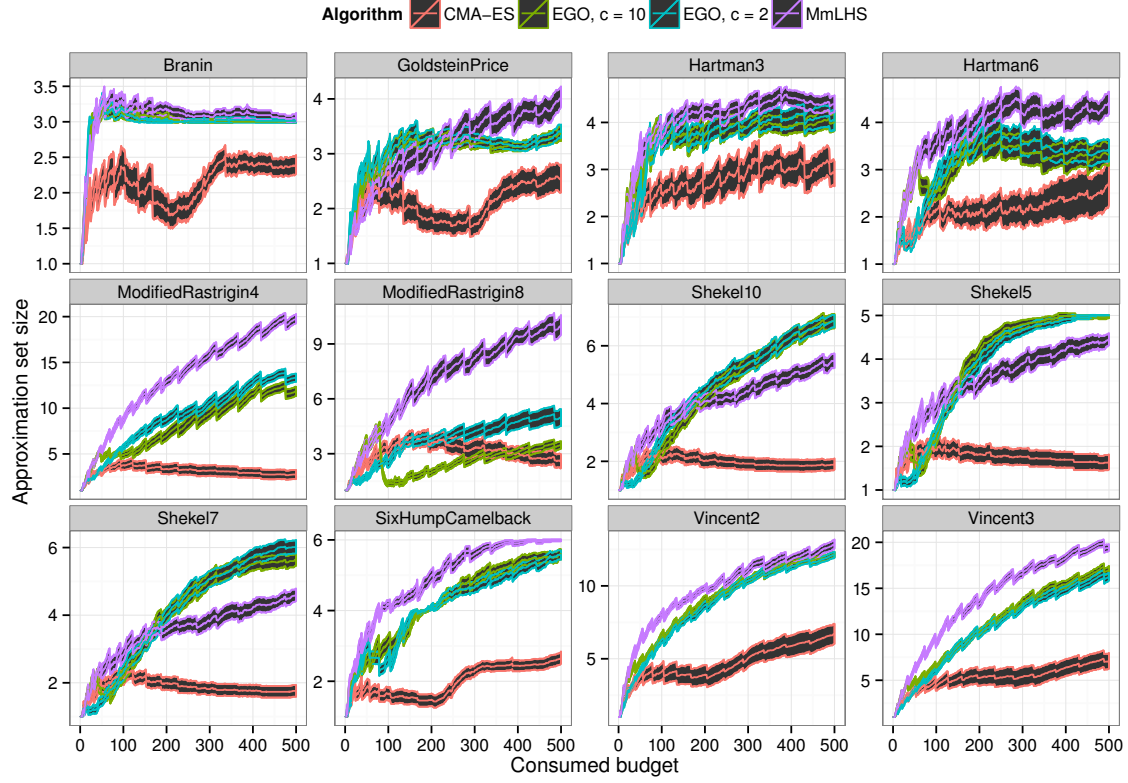


Figure 5: The number of basins identified by topographical selection after each evaluation.

approximation quality does not satisfy the PR criterion for all optima.

The running times of the algorithms in wall clock time naturally differ by orders of magnitude (see Fig. 6). The curve for MmLHS is slightly misleading, as the whole 500-point sample was computed in advance. So the curve begins with this cost and subsequently adds the cumulative cost of the test functions plus the cost for one topographical selection. In reality, MmLHS is of course always the cheapest algorithm in this setting, because one would not sample more points than one wants to evaluate.

Discussion: The results show that EGO represents an intermediate strategy between CMA-ES and MmLHS regarding the exploration-exploitation trade-off. It has the potential to detect several attraction basins with quite small budgets, but lacks the ability to approximate the corresponding optima with high precision. The f_{Δ} may stagnate for several hundred function evaluations. On the other hand, performance measures from multimodal optimization do often keep improving during this time.

The bends in the curves of CMA-ES in Fig. 2 are probably due to the aggressive stopping criteria, which prevent the algorithm from approximating the global optimum to a higher precision. This is the only explanation on problems as Branin or Vincent, which only contain global optima, and where we would expect a linear convergence behavior otherwise. However, this is not to be seen as a drawback, as we deliberately

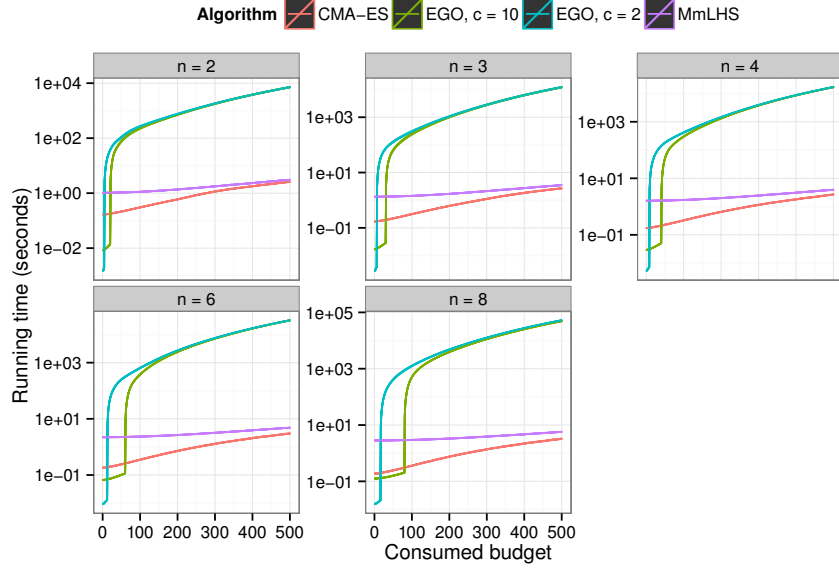


Figure 6: The wall clock time required by the algorithms.

chose this setting to obtain a better global search, and the restarts do clearly improve other measures (see Figs. 3 and 4). Tuning the initial step size might improve the CMA-ES performance slightly more. However, note that any improvement of CMA-ES would further strengthen the support for our hypothesis, so the omission is not critical.

5 Conclusions

We showed that *efficient global optimization* (EGO) is in fact not always the best algorithm for global optimization, i. e., the application it was originally designed for, except for extremely small budgets of approximately up to 200 function evaluations. By our experimental setup, this statement is restricted to rather low-dimensional ($2 \leq n \leq 8$), smooth, and generally well-behaved objective functions. However, as higher-dimensional, more multimodal, and/or non-continuous functions would pose even more difficulties to the meta-modelling, and other optimization algorithms as, e. g., CMA-ES are inherently more robust to such difficulties, because they use less assumptions about the problem, the statement might be extended to broader settings in the future. Of course more sophisticated sequential model-based optimization algorithms do already exist, and may not share some of the basic EGO’s weaknesses in global optimization, but our point is that EGO is actually fairly well-suited for the slightly different problem definition of finding multiple optima, when the optimization problem is expensive. Thus, combining it with a suitable basin identification heuristic makes it a strong competitor in this domain.

In future work, we shall look again at the basin identification mechanism and find better values or heuristics for the k parameter, or even another algorithm altogether.

Also, additional model-based optimization algorithms may be tested to find out if there exists an approach that is competitive in both the global and multi-local optimization case. Finally, the methods shall be thoroughly benchmarked on a larger set of problems in order to make stronger claims on their strengths and weaknesses.

References

- [1] Anne Auger and Nikolaus Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In *IEEE Congress on Evolutionary Computation (CEC)*, volume 2, pages 1777–1784, 2005.
- [2] Bernd Bischl, Simon Wessing, Nadja Bauer, Klaus Friedrichs, and Claus Weihs. Moi-mbo: Multiobjective infill for parallel model-based optimization. In *Learning and Intelligent Optimization, Lion 8*, pages 173–186. Springer, 2014.
- [3] Adam D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12:2879–2904, 2011.
- [4] Kalyanmoy Deb and Amit Saha. Multimodal optimization using a bi-objective evolutionary algorithm. *Evolutionary Computation*, 20(1):27–62, 2012.
- [5] Enrique del Castillo. *Process Optimization*, volume 105 of *International Series in Operations Research & Management Science*. Springer, 2007.
- [6] L.C.W. Dixon and G.P. Szegö. The global optimization problem: an introduction. In L.C.W. Dixon and G.P. Szegö, editors, *Towards Global Optimisation 2*, pages 1–15. North-Holland, Amsterdam, 1978.
- [7] Agoston E. Eiben and Mark Jelasity. A critical note on experimental research methodology in EC. In *IEEE Congress on Evolutionary Computation (CEC)*, volume 1, pages 582–587, 2002.
- [8] Paul Feliot, Julien Bect, and Emmanuel Vazquez. A Bayesian approach to constrained single- and multi-objective optimization. *Journal of Global Optimization*, 67(1):97–133, 2017.
- [9] Steffen Finck, Nikolaus Hansen, Raymond Ros, and Anne Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2009. Updated February 2010.
- [10] David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. In Yoel Tenne and Chi-Keong Goh, editors, *Computational Intelligence in Expensive Optimization Problems*, pages 131–162. Springer, 2010.
- [11] Donald R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

- [12] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.
- [13] Xiadong Li, Andries Engelbrecht, and Michael G. Epitropakis. Benchmark functions for CEC’2013 special session and competition on niching methods for multimodal function optimization. Technical report, RMIT University, Evolutionary Computation and Machine Learning Group, Australia, 2013.
- [14] Ian C. Parmee. The maintenance of search diversity for effective design space decomposition using cluster-oriented genetic algorithms (COGA) and multi-agent strategies (GAANT). In *Proceedings of 2nd International Conference on Adaptive Computing in Engineering Design and Control, ACEDC ’96*, pages 128–138. University of Plymouth, 1996.
- [15] Mike Preuss. Improved topological niching for real-valued global optimization. In *Applications of Evolutionary Computation*, volume 7248 of *Lecture Notes in Computer Science*, pages 386–395. Springer, 2012.
- [16] Mike Preuss. *Multimodal Optimization by Means of Evolutionary Algorithms*. Springer, 2015.
- [17] Mike Preuss, Günter Rudolph, and Simon Wessing. Tuning optimization algorithms for real-world problems by means of surrogate modeling. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO ’10*, pages 401–408. ACM, 2010.
- [18] Mike Preuss and Simon Wessing. Measuring multimodal optimization solution sets with a view to multiobjective techniques. In *EVOLVE – A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, volume 227 of *Advances in Intelligent Systems and Computing*, pages 123–137. Springer, 2013.
- [19] Luis Miguel Rios and Nikolaos V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.
- [20] Oliver Schütze, Xavier Esquivel, Adriana Lara, and Carlos A. Coello Coello. Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522, 2012.
- [21] András Sóbester, Stephen J. Leary, and Andy J. Keane. On the design of optimization strategies based on global response surface approximation models. *Journal of Global Optimization*, 33(1):31–59, 2005.
- [22] Ponnuthurai N. Suganthan, Nikolaus Hansen, Jing Jane Liang, Kalyanmoy Deb, Ying-Ping Chen, Anne Auger, and Santosh Tiwari. Problem definitions and eval-

uation criteria for the CEC 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University, Singapore, May 2005.

- [23] René Thomsen. Multimodal optimization using crowding-based differential evolution. In *IEEE Congress on Evolutionary Computation*, volume 2, pages 1382–1389, 2004.
- [24] Aimo Törn and Sami Viitanen. Topographical global optimization. In Christodoulos A. Floudas and Panos M. Pardalos, editors, *Recent Advances in Global Optimization*, Princeton Series in Computer Sciences, pages 384–398. Princeton University Press, 1992.
- [25] Aimo Törn and Antanas Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer, 1989.
- [26] Rasmus K. Ursem. Multinational evolutionary algorithms. In Peter J. Angeline, editor, *Proceedings of the Congress of Evolutionary Computation (CEC 99)*, volume 3, pages 1633–1640. IEEE Press, 1999.
- [27] Rasmus K. Ursem. From expected improvement to investment portfolio improvement: Spreading the risk in kriging-based optimization. In *Parallel Problem Solving from Nature – PPSN XIII: 13th International Conference, Proceedings*, pages 362–372. Springer, 2014.
- [28] Simon Wessing. *Two-stage methods for multimodal optimization*. PhD thesis, Technische Universität Dortmund, 2015. URL <http://hdl.handle.net/2003/34148>.
- [29] Simon Wessing, Günter Rudolph, and Mike Preuss. Assessing basin identification methods for locating multiple optima. In Panos M. Pardalos, Anatoly Zhigljavsky, and Julius Žilinskas, editors, *Advances in Stochastic and Deterministic Global Optimization*, pages 53–70. Springer, 2016.